Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	22621	(private or secret) near2 key	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:05
L2	24764	(public or shar$3) near2 key	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:05
L3	16113	1 and 2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:06
L4	82852	encrypt$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:06
L5	13728	3 and 4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:06
L6	1424534	ID or identifier or identification	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:07
L7	10552	5 and 6	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:12
L8	2	"6282656".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:29
L9	779658	send$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:30

L10	61808	receiv$3 near5 key	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:31
L11	18058	send$3 near5 key	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:30
L12	106682	recipient	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:32
L13	164237	10 or 12	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:30
L14	10203	11 and 13	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:30
L15	3501	14 and 7	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:30
L17	50181	receiv$3 adj3 (ID or identification or identifier)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:32
L18	51633	(recipient or receiv$3) adj3 (ID or identifier or identification)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:32
L19	50181	17 and 18	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:32
L20	817	15 and 19	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:32

read

L21	145	20 and @ad<"19980427"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/02/03 13:33

```
? show files
File  15:ABI/Inform(R) 1971-2006/Feb 03
         (c) 2006 ProQuest Info&Learning
File  16:Gale Group PROMT(R) 1990-2006/Feb 03
         (c) 2006 The Gale Group
File 148:Gale Group Trade & Industry DB 1976-2006/Feb 03
         (c)2006 The Gale Group
File 160:Gale Group PROMT(R) 1972-1989
         (c) 1999 The Gale Group
File 275:Gale Group Computer DB(TM) 1983-2006/Feb 03
         (c) 2006 The Gale Group
File 621:Gale Group New Prod.Annou.(R) 1985-2006/Feb 03
         (c) 2006 The Gale Group
File   9:Business & Industry(R) Jul/1994-2006/Feb 02
         (c) 2006  The Gale Group
File  20:Dialog Global Reporter 1997-2006/Feb 03
         (c) 2006 Dialog
File 476:Financial Times Fulltext 1982-2006/Feb 04
         (c) 2006 Financial Times Ltd
File 610:Business Wire 1999-2006/Feb 03
         (c) 2006 Business Wire.
File 613:PR Newswire 1999-2006/Feb 03
         (c) 2006 PR Newswire Association Inc
File 624:McGraw-Hill Publications 1985-2006/Feb 03
         (c) 2006 McGraw-Hill Co. Inc
File 634:San Jose Mercury  Jun 1985-2006/Feb 02
         (c) 2006 San Jose Mercury News
File 636:Gale Group Newsletter DB(TM) 1987-2006/Feb 03
         (c) 2006 The Gale Group
File 810:Business Wire 1986-1999/Feb 28
         (c) 1999 Business Wire
File 813:PR Newswire 1987-1999/Apr 30
         (c) 1999 PR Newswire Association Inc
File   2:INSPEC 1898-2006/Jan W2
         (c) 2006 Institution of Electrical Engineers
File  35:Dissertation Abs Online 1861-2006/Jan
         (c) 2006 ProQuest Info&Learning
File  65:Inside Conferences 1993-2006/Jan W5
         (c) 2006 BLDSC all rts. reserv.
File  99:Wilson Appl. Sci & Tech Abs 1983-2005/Dec
         (c) 2006 The HW Wilson Co.
File 256:TECINFOSOURCE 82-2005/DEC
         (c) 2006 INFO.SOURCES INC
File 474:New York Times Abs 1969-2006/Feb 02
         (c) 2006 The New York Times
File 475:Wall Street Journal Abs 1973-2006/Feb 02
         (c) 2006 The New York Times
File 583:Gale Group Globalbase(TM) 1986-2002/Dec 13
         (c) 2002 The Gale Group
? ds

Set     Items   Description
S1    7902848   KEY OR KEYS
S2    8448637   PRIVATE OR SECRET? ?
S3   32505035   PUBLIC OR SHAR???
S4      45272   S1 (3N) S2
S5     144308   S1(3N) S3
S6     349665   ENCRYPT???
S7       8676   S4 AND S5 AND S6
S8    1976616   ID OR IDS OR IDENTIFIER OR IDENTIFIERS OR IDENTIFICATION OR
```

```
                IDENTIFICATIONS
S9     314356   SENDER OR SENDER? ? OR ORIGINATOR? ?
S10   1092538   RECEIVER OR RECEIVERS OR RECIPIENT OR RECIPIENTS
S11      3118   S9 (5N) S1
S12      6123   S10 (5N) S1
S13       544   S7 AND S11 AND S12
S14      1948   S9 (3N) S8
S15      1358   S10 (3N) S8
S16         4   S13 AND S14 AND S15
S17         4   RD  (unique items)
?
PLEASE ENTER A COMMAND OR BE LOGGED OFF IN 5 MINUTES
?
```

read

? t s17/medium,k/1-4

17/K/1 (Item 1 from file: 15)
DIALOG(R)File 15:ABI/Inform(R)

00958306 96-07699
PGP: Pretty Good Privacy
Noor, Arshad
UNIX Review v13n2 PP: 31-38 Feb 1995
ISSN: 0742-3136 JRNL CODE: UXR
WORD COUNT: 3130

...ABSTRACT: companies wishing to use it for commercial purposes can obtain a commercial version of PGP. **Public - key** cryptography uses 2 **keys** : a **secret key** , called the **private key** , known only to the owner of the key; and a complementary **key** , called the **public key** , which is **public** information to the rest of the world. PGP combines the single- **key** and **public - key** cryptosystems. The process for getting PGP in the US is presented.
...TEXT: the person it claims to have come from? These two questions are the heart of **encryption** and authentication.

Encryption is the process of using a cryptographic scheme to convert a source message into incomprehensible...

...came from the person it claims to have come from. Here's how PGP handles **encryption** and authentication.

Encryption

The beginning of any **encryption** process is the source file, called a plaintext file. This file may be an ASCII...

...point of view, it is an unencrypted plain-text file. The end result of the **encryption** process is an **encrypted** file called the ciphertext file. Plaintext files are converted to ciphertext files using a cryptographic key. In what is known as traditional, single- **key** , or **secret - key** cryptographic schemes, a single key, which is not meant to be shared with anyone else, is used to **encrypt** a file and decrypt it back to its original state. The **secret - key** mechanism is used in the Data **Encryption** Standard (DES), in the UNIX crypt command, and in Kerberos.

This method may be satisfactory for simple uses of privacy, such as an individual user **encrypting** files on the hard disk or on a private network where the chances of attack...

...minimal. But when two or more users need to share messages with each other over **public** networks, **secret - key** schemes fail because they require that the **encryption key** be **shared** between the parties. If the channel through which they **share keys** is not secure, then the whole purpose of **encryption** is defeated.

Recognizing the problem of **sharing secret keys** over **public** channels, Whitfield Diffie and Martin Hellman invented the **public - key** cryptographic scheme in 1976. **Public - key** cryptography uses two **keys** : **a secret key , called the private key , is known only to the owner of**

the key; and a complementary **key**, called the **public key**, is **public** information to the rest of the world. The two keys are mathematically related so that when a message is **encrypted** using the **public key** of an individual, only the complementary **private key** can decrypt the ciphertext. Thus, the **public key** can now be **shared** across the globe over insecure channels with the assurance that without the **private key**, a secure message has extremely low probability of being compromised. This is the cryptographic scheme used in Solaris NIS+.

The only problem with **public - key** cryptography is that, depending on the size of the chosen key and the size of the plaintext, **encrypting** and decrypting messages can take a very long time. To ensure that even PCs can use this technology in a realistic manner, PGP combines the single- **key** and **public - key** cryptosystems to create an effective compromise. The **public - key** technology used by PGP is the well-known RSA (Rivest, Shamir, and Adleman) **public - key** technology. Even though this technology is patented in the United States, through a consortium of **public - key** vendors called **Public Key** Partners (PKP), RSA has placed the RSAREF technology in the public domain for noncommercial uses only.

When a plaintext file is ready to be **encrypted**, PGP goes through the following steps to arrive at the ciphertext:

* PGP first compresses the...

...compression software.

* PGP then generates a random session key, using an algorithm called International Data **Encryption** Algorithm (IDEA).

* Using this session key, PGP **encrypts** the compressed file to create the ciphertext file.

* The session key is then **encrypted** using the **public key** of the **recipient** and prefixed to the front ...PGP goes through the following steps to arrive at the decrypted plaintext:
* Recognizing that a **public key** is in the ciphertext file, PGP prompts the user for the **private - key** password, or pass-phrase, of the holder of that **public key** (usually the **recipient**).

* Upon receiving the correct pass-phrase, PGP uses the **private key** to decrypt the session key.

* Using the session key, PGP now decrypts the ciphertext file...

...25MHz PC, this article, which is more than 3,000 words, took 5 seconds to **encrypt**.

Authentication

Authentication attempts to establish that the message received has not been altered and that...

...digital signature is a mathematically computed value that depends on the plaintext file and the **private key** of the **sender**. Thus, every unique message will have a different digital signature. Any alteration of either item...

...algorithm, which is in the public domain for noncommercial use.

* The message digest is then **encrypted** using the **private key** of the **sender** . A time stamp is added to the computed value to create the digital signature, which...

...the authenticity of the message:

* Recognizing a digital signature in the document, PGP uses the **public key** of the **sender** to decrypt the message digest from the ciphertext file.

* It then computes another message digest...popular time-zone values.

After you have set up these variables, you should create your **public** and **private key** . The command for this is:

pgp -kg

where the -k option indicates a key-management...

...want to generate a key.
You will be prompted for the desired size of your **public key** . The choices are 512 bits for low commercial grade, 768 bits for high commercial grade...

...better. Otherwise, you'll spend a fair amount of idle time as your hardware frequently **encrypts** and decrypts messages. The larger the key, the longer it takes.

After you've entered...

...mail address or are known by more than one name, you can later edit your **public key** and add that information. This allows you to be known through different aliases yet have only one **public key** .

After you've entered your ID, you will be prompted for a pass phrase. PGP ...

...PGP uses this text to create a random seed for generating the session key to **encrypt** your messages. The intervals of time between your keystrokes are used to generate the random...

...not the text itself. This random seed is updated every time you use PGP to **encrypt** messages, which reduces the probability of someone discovering the session key.

After the required keystrokes...

...DOS; in UNIX they are usually in $HOME/.pgp): pubring.pgp and secring.pgp, the **public key** ring and the **private** (**secret**) **key** ring files, respectively. To test PGP, generate another key for a fictitious user, such as John Doe.

To **encrypt** a file, such as a copy of/etc/motd called $HOME/mymotd to be mailed...

...John Doe, type the following command:

pgp -e mymotd doe

where the -e option means **encrypt** , the filename is mymotd, and the recipient is any user in the pubring.pgp file...

...save it under another filename.
Upon reading mymotd.pgp, PGP recognizes it to be an **encrypted** file, extracts the **recipient** 's user **ID** from the file, and prompts for the secret pass-phrase for that user ID. After...

...id

where the -s option tells PGP to sign the document after **encrypting** it (because of the -e option), and the -u option indicates the user ID that...

...be used to sign the document (usually your own). You will be prompted for your **private key** pass-phrase to create the signature. Upon receipt, the recipient types in the same command...

...file is decrypted correctly, PGP recognizes the digital signature attached to it and uses the **public key** of the **sender** to authenticate this signature. If everything is correct, you will be told that the message emanated from that user, and PGP will display the full user **ID** of the **sender** .

PGP offers myriad options for various forms of **encryption** , decryption, authentication, and key management. The config.txt file allows you to customize options for...

17/K/2 (Item 2 from file: 15)
DIALOG(R)File 15:ABI/Inform(R)

00747520 93-96741
Internet Privacy Enhanced Mail
Kent, Stephen T
Communications of the ACM v36n8 PP: 48-60 Aug 1993
ISSN: 0001-0782 JRNL CODE: ACM
WORD COUNT: 9143

...ABSTRACT: makes use of a variety of cryptographic algorithms. These algorithms provide for message integrity, message **encryption** , and distribution of the cyptographic keys used to encipher messages. If **public - key** cryptoalgorithms are employed for key management, then additional algorithms must be specified. The base PEM...
...TEXT: is replicated in the PEM filter. For example, the user might be required to supply **recipient identifiers** twice, once for PEM processing and once for email addressing. In Figure 1 the message...

...cryptographic algorithms (see "Cryptographic Concepts and Terminology" sidebar). These algorithms provide for message integrity, message **encryption** , and distribution of the cryptographic keys used to encipher messages. If **public - key** cryptoalgorithms are employed for key management, then additiona! algorithms must be specified.

The base PEM...

...In addition to the specification of message-processing facilities, the PEM standards provide for a **public - key** certification infrastructure. Although PEM allows for the use of either **secret - key** or **public - key** crypto-algorithms for key distribution, the standards encourage the use of **public - key** cryptography because of its ability to support a very large, distributed user community. The specific approach to **public - key** cryptography adopted for PEM is based on the use of certificates, as defined in CCITT...

...4] and as adopted by ISO for both directory and messaging security (see sidebar on **Public - Key** Certificates).

The PEM standards establish a specific framework for a **public key** certification system for several reasons. Although PEM makes use of X.509 certificates, the international...

...and deemed sensitive, might be omitted or a benign Subject might be substituted (e.g., " **Encrypted** Message"). A sensitive Subject field can be enclosed within the PEM-protected content, affording it confidentiality. Only one portion of the header data, namely **recipient identifiers** (e.g., mailbox addresses), is required to control PEM processing. If the (optional) confidentiality service is selected by the message **originator** , these **recipient identifiers** are used to control message encipherment.

In Internet email, the header data is separated from...provide different combinations of security services for different messaging contexts: MIC-CLEAR, MIC-ONLY, and **ENCRYPTED** .[4] A MIC-CLEAR message employs a cryptographic message integrity code (MIC) to check the...

...without being transformed in a fashion that would invalidate the integrity and authenticity checks. An **ENCRYPTED** message adds the confidentiality service to integrity and authenticity. This message type also uses the encoding transformation described for MIC-ONLY, since otherwise the (binary) output of the **encryption** processing would be unable to transit many email systems (which were designed to transfer text ...

...three major transformation steps: canonicalization, computation of the message integrity code (MIC) and optional message **encryption** , followed by optional 6-bit encoding. These steps are illustrated in Figure 3. Since the **encryption** and encoding constitute optional processing steps, a field indicating the PEM processing options (Proc-Type...

...transformation steps are later assembled to form the PEM header. Figure 4 contains a sample **ENCRYPTED** PEM messae to illustrate the elements of a PEM header.

CANONICALIZATION

The first step in asymmetric cryptography, the MIC is protected using the private component of the **originator** 's **public - key** pair. This effects a digital signature on the message, which can be verified by any...

...MIC value and also specifies the means used to protect the MIC (e.g., RSA **encryption** is employed in Figure 4).

To facilitate the ability of a recipient to establish the...

...identification when using asymmetric cryptography (i.e., the Originator-Certificate field). This field conveys the **public - key** certificate of the **originator** , which will be used by a recipient in verifying the integrity of the MIC value...

...in the context of delivery processing.

Finally, the transformation of the MIC value using the **originator** 's **public key** does not protect against disclosure of this value. It is not possible to work backward...

...MIC value to determine the content of a message; thus even if a message is **encrypted** to provide confidentiality one might not feel a need to **encrypt** the MIC value. However, an attacker might make educated guesses about the message content and...

...guesses against the (signed) MIC value in the PEM header. Therefore, if the message is **encrypted** using PEM, the MIC value is also **encrypted** (using the same key employed to **encrypt** the message content) to protect against this attack. Also, since the value of the MIC (before or after **encryption**) is usually binary, it may not be possible to transmit it using a messaging system...

...6-bit encoding is employed for this field as is applied to the message content).

ENCRYPTION

The second PEM processing step also provides message **encryption** , if selected by the originator. This processing is performed only if the PEM header specifies a Proc-Type value of " **ENCRYPTED** ." Any padding required by the message **encryption** algorithm is applied to the canonicalized plaintext before **encryptio**n .[5] A message encipherment key, to be used exclusively to **encrypt** this ones message, is generated by the originator. The data **encryption** algorithm employed in PEM, and its mode of use, is not fixed but is another parameter, specified in the DEK-Infofield of the PEM header. If the **encryption** algorithm requires any parameters, these are also specified in this field. The canonical (padded as required) message text is then **encrypted** using the per-message key. The example PEM message shown in Figure 4 uses the Data **Encryption** Standard (DES) [9] in cipher block chaining (CBC) mode [7] for encipherment. This mode of...

...Info in the sample message.

As described in the preceding paragraphs, a PEM message is **encrypted** exactly once, irrespective of the number of recipients. Only one copy of this **encrypted** message is submitted to the message transfer system, and copies of this message are delivered...

...user mailboxes just as is done for regular, non-PEM messages. Effecting this multicast message **encryption** capability requires a key distribution technique that differs from those commonly employed for point-to...

...communication [12]. Using asymmetric cryptography for key distribution, one copy of the message key is **encrypted** using the public component of

the **public - key** pair for each **recipient** .[6] In this way, each copy of the message key is protected in a fashion that makes it decipherable by exactly one **recipient** .

Each **encrypted** message **key** copy is placed in a Key-Info field, following an identifier for the **public - key** algorithm used to **encrypt** the copy of the message key. Each Key-Info field is preceded by a **Recipient - ID** -Asymmetric field that identifies the recipient, by the X.500 distinguished name of his certificate...

...header fields provides the information required for a recipient to decrypt a message. If different **recipients** employ different **key** distribution algorithms, this is naturally accommodated by this pairing of per-recipient fields. In Figure 4, RSA is employed as the **public - key encryption** algorithm and it is identified in the Key-Info field.

ENCODING FOR TRANSMISSION

The third (final) processing step renders **ENCRYPTED** or MIC-ONLY message into a character set suitable for transmission through a messaging system ...

...can vary for different messaging system environments. The encoding step initially specified transforms the (optionally **encrypted**) message text into a restricted 6-bit alphabet, plus linelength constraints, that make the encoding...

...email gateways that link the Internet to other messaging systems. If the message has been **encrypted** , this encoding serves to transform the 8-bit (binary) ciphertext into a form that can...

...to any portion of the third processing step.

Even if the message has not been **encrypted** , this encoding step ensures, with high probability, that the canonicalized version of the message (produced...be performed by the recipient. Our sample message uses version 4 of PEM and is **ENCRYPTED** .

DECODING AND DECRYPTION

For a message of type **ENCRYPTED** or MIC-ONLY, the first step is the inversion of any encoding step applied by...

...form). The decoding performed is determined by the message system context.

If the message is **ENCRYPTED** , the recipient scans the PEM header to locate the the **Recipient - ID** -Asymmetric field that uniquely identifies him. The **recipient** then examines the **Key** -Info field immediately following this ID field. The first parameter of this field specifies the...
...key.[8] In the asymmetric cryptographic context, the recipient uses the private component of his **public - key** pair to decrypt the second field, yielding the message key.

The DEK-Info field, which...

...and any parameters necessary for decryption. In the sample message, this field specifies that each **recipient** would use the decrypted message **key**

in conjunction with the DES, in CBC mode, with the initialization vector included in the DEK-Info field. The **recipient** can now use the message **key** , as indicated by the Key-Info field, to decrypt the message text. After decryption, the...

...component, in support of originator authentication.

In principle, this identification requires validating a sequence of **public - key** certificates that terminates with the certificate of the originator. As noted in a previous subsection...

...but decrypted) form, along with the PEM header fields needed for signature verification (MIC-Info, **Originator - ID** -Asymmetric, **Originator** -Certificate, IssuerCertificate). This form of storage is appropriate if the user wishes to forward a...

...modification of the message while in storage. Finally, the user may save the message in **encrypted** form, to additionally protect the message against disclosure while in storage.

THE INTERNET **PUBLIC - KEY** CERTIFICATION SYSTEM

The PEM specifications encourage use of **public - key** cryptography for message integrity, **originator** authentication, and for distribution of data **encryption** keys. As noted previously, PEM makes use of **public - key** certificates that conform to CCITT X.509 (see the **Public Key** Certificates sidebar). The X.509 recommendation defines an authentication framework, not only a certificate format...policies (e.g., they may strive for varying degrees of assurance in vouching for name- **public - key** bindings). However, X.509 makes no provisions for users to learn what policy each CA...

...imposed by these PCAs. MIT would achieve this capability by having two certificates, with different **public keys** , each signed by the relevant PCA.
In addition to the organizational CAs shown for MIT...framework for the discussion that follows.

A cryptoalgorithm is used to transform data through an **encryption** process. The input to this process is called plaintext and the output is called ciphertext, The **encryption** is inverted by a decryption process, which accepts ciphertext as input and yields plaintext. In...

...in controlled by a key, which is a parameter to the process. In a symmetric (**secret - key**) cipher, the same key is used to **encrypt** and decrypt data. (In Figure A. KEY-1 and KEY-2 would be identical.) That **key** is kept **secret** and is shared by a transmitter and a receiver. Symmetric ciphers typically exhibit good performance and are used to **encrypt** user data. Privacy Enhanced Mail uses symetric ciphers to **encrypt** messages.

In an assymetric (**public - key**) cryptoalgorithm a pair of distinct, but mathematically related, keys are used for **encryption** and decryption. One **key** is kept **private** and is known only to its owner, whereas the other key is made publicly known, hence the term, " **public - key** cryptography." (In Figure A, KEY-1 and KEY-2 would be distinct and either could be the **public** or **private key** .) Data **encrypted** with a user's **private key** can be decrypted using his **public key** , and vice versa, in the general

model of **public - key** cryptography. Since the performance of asymmetric ciphers generally is not as good as that of...

...ciphers, the former usually are not used to encipher data directly. Instead, the asymmetry of **public - key** is often exploited to distribute symmetric keys and for digital signatures.

A digital signature is often effected using **public - key** cryptography and a (one-way) hash function, as illustrated in Figure B. The hash function...

...hash value, making it a "fingerprint" of the data. A user digitally signs data by **encrypting** the hash value of the data using his private component. Using an algorithm such as...

...the results. A match indicates a valid digital signature.

Privacy Enhanced Mail makes use of **public - key** ciphers to **encrypt** the symmetric key used to **encrypt** a message and to digitally sign a message.

PUBLIC KEY CERTIFICATES: THE X.509 WAY

A **public - key** certificate is a data structure used to securely bind a **public key** to attributes. The attributes can consist of identification information, for example, a name, or authorization...

...certificates is contained within the international standards for directories. An X.509 certificate binds a **public key** to a directory name and identifies the entity who vouches for the binding. The whole...

...name of the entity that vouches for the binding between the subject name and the **public key** contained in the certificate (see the following subsection on distinguished names). The subject and issuer...

...certificate is valid, in much the same way as many credit cards are marked. The **public key** alluded to earlier, along with an identifier to specify the algorithm and any parameters required...

...the digital signature applied to the certificate. Both a one-way hash function and a **public - key** signature algorithm will be specified (e.g., the RSA **public key** algorithm and the MD2 hash algorithm in this example). This signature is applied by the...

...section. "Cryptographic Concepts and Terminology." This use of certificates transforms the problem of acquiring the **public key** associated with a user into one of acquiring the **public key** of the issuer of the user's certificate. This issuer also will have a certificate ...

...The validation algorithm must conclude at some point, however, implying that the user holds a **public key** obtained through some out-of-band, integrity-secure channel (not through an untrusted network).

A...revoked ("hot-listed") by the issuer if the binding between the subject and the subject **public key** is no longer valid. This revocation of the binding may result from a number of...

...end users, and thus is not addressed in this article.

5. Any padding applied for **encryption** is removed as part of the decryption process performed by each recipient, so the padding...

...is employed for key distribution, the same general approach is employed. A different, symmetric cryptograph **key** is **shared** by each **originator - recipient** pair and that **key** is used to **encrypt** the message **key** on a per- **recipient** basis.
7. The use of the recipient's distinguished name, rather than a mailbox name...

...a single cryptographically authenticated identity.

8. This second field will generally be encoded, since the **encrypted** kev is a binary value, and thus must be decoded before it can be decrypted... Data Authentication, May 1985.

9. Federal information processing standards publication (FIPS PUB) 46-1, Data **encryption** standard, Reaffirmed Jan. 1988 (supersedes FIPS PUB 46, Jan. 1977).

10. Information processing systems--Open...

...computers and computer networks, protocols for secure email and directory services, and the establishment of **public - key** certification systems to provide the necessary infrastructure for secure applications. Author's Present Address: Bolt...

17/K/3 (Item 1 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01770774 SUPPLIER NUMBER: 16652259 (USE FORMAT 7 OR 9 FOR FULL TEXT)
PGP: Pretty Good Privacy. (a freeware Internet e-mail security program)(Cover Story)
Noor, Arshad
UNIX Review, v00000013, n2, p31(6)
Feb, 1995
DOCUMENT TYPE: Cover Story ISSN: 0742-3136 LANGUAGE: ENGLISH
RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 3724 LINE COUNT: 00283

...ABSTRACT: individuals and companies wanting to use it for commercial purposes can get a commercial version. **Encryption** uses a cryptographic scheme to convert a source message into incomprehensible patterns, while authentication ensures...
... the person it claims to have come from? These two questions are the heart of **encryption** and authentication.
Encryption is the process of using a cryptographic scheme to convert a source message into incomprehensible...

...came from the person it claims to have come from. Here's how PGP handles **encryption** and authentication.

Encryption

The beginning of any **encryption** process is the source file, called a plaintext file. This file may be an ASCII...

...cryptographic point of view, it is an unencrypted plaintext file. The

end result of the **encryption** process is an **encrypted** file called the ciphertext file. Plaintext files are converted to ciphertext files using a cryptographic key. In what is known as traditional, single- **key** , or **secret** - **key** cryptographic schemes, a single key, which is not meant to be shared with anyone else, is used to **encrypt** a file and decrypt it back to its original state. The **secret** - **key** mechanism is used in the Data **Encryption** Standard (DES), in the UNIX crypt command, and in Kerberos.

This method may be satisfactory for simple uses of privacy, such as an individual user **encrypting** files on the hard disk or on a private network where the chances of attack...

...minimal. But when two or more users need to share messages with each other over **public** networks, **secret** - **key** schemes fail because they require that the **encryption** **key** be **shared** between the parties. If the channel through which they **share** **keys** is not secure, then the whole purpose of **encryption** is defeated.

Recognizing the problem of **sharing** **secret** **keys** over **public** channels, Whitfield Diffie and Martin Hellman invented the **public** - **key** cryptographic scheme in 1976. **Public** - **key** cryptography uses two **keys** : a **secret** **key** , called the **private** **key** , is known only to the owner of the key, and a complementary **key** , called the **public** **key** , is **public** information to the rest of the world. The two keys are mathematically related so that when a message is **encrypted** using the **public** **key** of an individual, only the complementary **private** **key** can decrypt the ciphertext. Thus, the **public** **key** can now be **shared** across the globe over insecure channels with the assurance that without the **private** **key** , a secure message has extremely low probability of being compromised. This is the cryptographic scheme used in Solaris NIS+.

The only problem with **public** - **key** cryptography is that, depending on the size of the chosen key and the size of the plaintext, **encrypting** and decrypting messages can take a very long time. To ensure that even PCs can use this technology in a realistic manner, PGP combines the single- **key** and **public** - **key** cryptosystems to create an effective compromise. The **public** - **key** technology used by PGP is the well-known RSA (Rivest, Shamir, and Adleman) **public** - **key** technology. Even though this technology is patented in the United States, through a consortium of **public** - **key** vendors called **Public** **Key** Partners (PKP), RSA has placed the RSAREF technology in the public domain for noncommercial uses only.

When a plaintext file is ready to be **encrypted** , PGP goes through the following steps to arrive at the ciphertext:

* PGP first compresses the...

...compression software.

* PGP then generates a random session key, using an algorithm called International Data **Encryption** Algorithm (IDEA).

* Using this session key, PGP **encrypts** the compressed file to create the ciphertext file.

* The session key is then **encrypted** using the **public** **key** of the **recipient** and prefixed to the front of the ciphertext file.

* The ciphertext file is now ready...

...PGP goes through the following steps to arrive at the decrypted plaintext:

* Recognizing that a **public** **key** is in the ciphertext file, PGP prompts the user for the **private** - **key** password, or pass-phrase, of the holder of that **public** **key** (usually the **recipient**).

* Upon receiving the correct pass-phrase, PGP uses the **private** **key**

to decrypt the session key.
* Using the session key, PGP now decrypts the ciphertext file...

...25Mhz PC, this article, which is more than 3,000 words, took 5 seconds to **encrypt** .

Authentication

Authentication attempts to establish that the message received has not been altered and that...

...digital signature is a mathematically computed value that depends on the plaintext file and the **private key** of the **sender** . Thus, every unique message will have a different digital signature. Any alteration of either item...

...algorithm, which is in the public domain for noncommercial use.
* The message digest is then **encrypted** using the **private key** of the **sender** . A time stamp is added to the computed value to create the digital signature, which...

...the authenticity of the message:
* Recognizing a digital signature in the document, PGP uses the **public key** of the **sender** to decrypt the message digest from the ciphertext file.
* It then computes another message digest...popular time-zone values.

After you have set up these variables, you should create your **public** and **private key** . The command for this is:

pgp -kg

where the -k option indicates a key-management...

...want to generate a key.

You will be prompted for the desired size of your **public key** . The choices are 512 bits for low commercial grade, 768 bits for high commercial grade...

...better. Otherwise, you'll spend a fair amount of idle time as your hardware frequently **encrypts** and decrypts messages. The larger the key, the longer it takes.

After you've entered...

...mail address or are known by more than one name, you can later edit your **public key** and add that information. This allows you to be known through different aliases yet have only one **public key** .

After you've entered your ID, you will be prompted for a pass-phrase. PGP...

...PGP uses this text to create a random seed for generating the session key to **encrypt** your messages. The intervals of time between your keystrokes are used to generate the random...

...not the text itself. This random seed is updated every time you use PGP to **encrypt** messages, which reduces the probability of someone discovering the session key.

After the required keystrokes...

...DOS; in UNIX they are usually in $HOME/.pgp): pubring.pgp and secring.pgp, the **public key** ring and the **private** (**secret**) **key** ring files, respectively. To test PGP, generate another key for a fictitious user, such as John-Doe.

To **encrypt** a file, such as a copy of/etc/motd called $HOME/my-motd to be...

...John Doe, type the following command:
pgp -e mymotd doe
where the -e option means **encrypt** , the filename is mymotd, and the recipient is any user in the pubring.pgp file...

...save it under another filename.
Upon reading mymotd.pgp, PGP recognizes it to be an **encrypted** file, extracts the **recipient** 's user **ID** from the file, and prompts for the secret pass-phrase for that user ID. After...

...id
where the -s option tells PGP to sign the document after **encrypting** it (because of the -e option), and the -u option indicates the user ID that ...

...be used to sign the document (usually your own). You will be prompted for your **private key** pass-phrase to create the signature. Upon receipt, the recipient types in the same command...

...file is decrypted correctly, PGP recognizes the digital signature attached to it and uses the **public key** of the **sender** to authenticate this signature. if everything is correct, you will be told that the message emanated from that user, and PGP will display the full user **ID** of the **sender** .

PGP offers myriad options for various forms of **encryption** , decryption, authentication, and key management. The config.txt file allows you to customize ...h Displays all the following options with a brief explanation of what they do. -e **Encrypts** a plaintext file with the receipt's **public key** . -s Signs plaintext file with your **secret key** . -a Stores the ciphertext file in ASCII format do that it can be mailed using

SMTP. -c **Encrypts** a plaintext file with conventional cryptography. -o Names the output file after decryption. -t Implements...

...ID for digital signing (usually your own).
Key Management Options
-kg Generates your own unique **public** / **secret key** pair. -ka Adds a **public** or (**secret** - **key** file's contents) to your **public** or **secret key**

ring. -kx Extracts (copies) a **key** from your public or **secret** - **key** ring for possible

mailing to a colleague or friend with whom you will exchange messages. -kv Displays the contents of your **public** - **key** ring. -kvc Displays the "fingerprint" of a public to help verify it over the telephone

with it's owner. -ke Edits the user ID or pass-phrase for your **secret key** . -kr Removes a key or a user ID from your **public** - **key** ring. -ks Signs and certifies someone else's **public key** on your **public** - **key** ring. -kd Permanently revokes your own key, issuing a key compromise certificate

that indicates the other **public** - **key** rings should not use your **public key** ;

also disables or re-enables a **public key** on your **public** - **key** ring.

Miscellaneous Options

-d Decrypts a message and leaves the signature on it intact. -sb...

17/K/4 (Item 2 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01615372 SUPPLIER NUMBER: 14207369 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Internet Privacy Enhanced Mail. (development of security standards for Internet computer network) (includes related articles on cryptography, on the X.509 standard and on distinguished names)
Kent, Stephen T.
Communications of the ACM, v36, n8, p48(13)
August, 1993
ISSN: 0001-0782 LANGUAGE: ENGLISH RECORD TYPE: FULLTEXT; ABSTRACT
WORD COUNT: 9696 LINE COUNT: 00785

... is replicated in the PEM filter. For example, the user might be required to supply **recipient identifiers** twice, once for PEM processing and once for email addressing. In Figure 1 the message...cryptographic algorithms (see "Cryptographic Concepts and Terminology" sidebar). These algorithms provide for message integrity, message **encryption** , and distribution of the cryptographic keys used to encipher messages. If **public - key** cryptoalgorithms are employed for key management, then additional algorithms must be specified.
The base PEM...

...In addition to the specification of message-processing facilities, the PEM standards provide for a **public - key** certification infrastructure. Although PEM allows for the use of either **secret - key** or **public - key** cryptoalgorithms for **key** distribution, the standards encourage the use of **public - key** cryptography because of its ability to support a very large, distributed user community. The specific approach to **public - key** cryptography adopted for PEM is based on the use of certificates, as defined in CCITT...

...4] and as adopted by ISO for both directory and messaging security (see sidebar on **Public - Key** Certificates).
The PEM standards establish a specific framework for a **public key** certification system for several reasons. Although PEM makes use of X.509 certificates, the international...

...and deemed sensitive, might be omitted or a benign Subject might be substituted (e.g., " **Encrypted** Message"). A sensitive Subject field can be enclosed within the PEM-protected content, affording it confidentiality. Only one portion of the header data, namely **recipient identifiers** (e.g., mailbox addresses), is required to control PEM processing. If the (optional) confidentiality service is selected by the message **originator** , these **recipient identifiers** are used to control message encipherment.
In Internet email, the header data is separated from...
...provide different combinations of security services for different messaging contexts: MIC-CLEAR, MIC-ONLY, and **ENCRYPTED** .(4) A MIC-CLEAR message employs a cryptographic message integrity code (MIC) to check the ...without being transformed in a fashion that would invalidate the integrity and authenticity checks. An **ENCRYPTED** message adds the confidentiality service to integrity and authenticity. This message type also uses the encoding transformation described for MIC-ONLY, since otherwise the (binary) output of the **encryption** processing would be

unable to transit many email systems (which were designed to transfer text ...

...three major transformation steps: canonicalization, computation of the message integrity code (MIC) and optional message **encryption** , followed by optional 6-bit encoding. These steps are illustrated in Figure 3. Since the **encryption** and encoding constitute optional processing steps, a field indicating the PEM processing options (Proc-Type...

...transformation steps are later assembled to form the PEM header. Figure 4 contains a sample **ENCRYPTED** PEM message to illustrate the elements of a PEM header.

Canonicalization

The first step in...

...4, which uses asymmetric cryptography, the MIC is protected using the private component of the **originator** 's **public** - **key** pair. This effects a digital signature on the message, which can be verified by any...

...MIC value and also specifies the means used to protect the MIC (e.g., RSA **encryption** is employed in Figure 4).

To facilitate the ability of a recipient to establish the...

...identification when using asymmetric cryptography (i.e., the Originator-Certificate field). This field conveys the **public** - **key** certificate of the **originator** , which will be used by a recipient in verifying the integrity of the MIC value...in the context of delivery processing.

Finally, the tranformation of the MIC value using the **originator** 's **public** **key** does not protect against disclosure of this value. It is not possible to work backward...

...MIC value to determine the content of a message; thus even if a message is **encrypted** to provide confidentiality one might not feel a need to **encrypt** the MIC value. However, an attacker might make educated guesses about the message content and...

...guesses against the (signed) MIC value in the PEM header. Therefore, if the message is **encrypted** using PEM, the MIC value is also **encrypted** (using the same key employed to **encrypt** the message content) to protect against this attack. Also, since the value of the MIC (before or after **encryption**) is usually binary, it may not be possible to transmit it using a messaging system...

...6-bit encoding is employed for this field as is applied to the message content).

Encryption

The second PEM processing step also provides message **encryption** , if selected by the originator. This processing is performed only if the PEM header specifies a Proc-Type value of " **ENCRYPTED** ." Any padding required by the message **encryption** algorithm is applied to the canonicalized plaintext before **encryption** .(5) A message encipherment key, to be used exclusively to **encrypt** this one message, is generated by the originator. The data **encryption** algorithm employed in PEM, and its mode of use, is not fixed but is another parameter, specified in the DEK-Info field of the PEM header. If the **encryption** algorithm requires any parameters, these are also specified in this field. The canonical (padded as required) message text is then **encrypted** using the per-message key. The example PEM message shown in Figure 4 uses the Data **Encryption** Standard (DES) [9] in

cipher block chaining (CBC) mode [7] for encipherment. This mode of...

...Info in the sample message.
As described in the preceding paragraphs, a PEM message is **encrypted** exactly once, irrespective of the number of recipients. Only one copy of this **encrypted** message is submitted to the message transfer system, and copies of this message are delivered...

...user mailboxes just as is done for regular, non-PEM messages. Effecting this multicast message **encryption** capability requires a key distribution technique that differs from those commonly employed for point-to...

...communication [12]. Using asymmetric cryptography for key distribution, one copy of the message key is **encrypted** using the public component of the **public - key** pair for each **recipient** .(6) In this way, each copy of the message key is protected in a fashion that makes it decipherable by exactly one **recipient** .

Each **encrypted** message **key** copy is placed in a Key-Info field, following an identifier for the **public - key** algorithm used to **encrypt** the copy of the message key. Each Key-Info field is preceded by a **Recipient - ID** -Asymmetric field that identifies the recipient, by the X.500 distinguished name of his certificate...
...header fields provides the information required for a recipient to decrypt a message. If different **recipients** employ different **key** distribution algorithms, this is naturally accommodated by this pairing of per-recipient fields. In Figure 4, RSA is employed as the **public - key encryption** algorithm and it is identified in the Key-Info field.

Encoding for Transmission

The third (final) processing step renders **ENCRYPTED** or MIC-ONLY message into a character set suitable for transmission through a messaging system...

...can vary for different messaging system environments. The encoding step initially specified transforms the (optionally **encrypted**) message text into a restricted 6-bit alphabet, plus line-length constraints, that make the...

...email gateways that link the Internet to other messaging systems. If the message has been **encrypted** , this encoding serves to transform the 8-bit (binary) ciphertext into a form that can...

...to any portion of the third processing step.
Even if the message has not been **encrypted** , this encoding step ensures, with high probability, that the canonicalized version of the message (produced...

...be performed by the recipient. Our sample message uses version 4 of PEM and is **ENCRYPTED** .

Decoding and Decryption

For a message of type **ENCRYPTED** or MIC-ONLY, the first step is the inversion of any encoding step applied by...

...form). The decoding performed is determined by the message system context.

If the message is **ENCRYPTED** , the recipient scans the PEM header to locate the **Recipient - ID** -Asymmetric field that uniquely identifies him. The **recipient** then examines the **Key** -Info field immediately following this ID field. The first parameter of this field specifies the...

...key.(8) In the asymmetric cryptographic context, the recipient uses the private component of his **public - key** pair to decrypt the second field, yielding the message key.

The DEK-Info field, which...

...and any parameters necessary for decryption. In the sample message, this field specifies that each **recipient** would use the decrypted message **key** in conjunction with the DES, in CBC mode, with the initialization vector included in the DEK-Info field. The **recipient** can now use the message **key** , as indicated by the Key-Info field, to decrypt the message text. After decryption, the...

...component, in support of originator authentication.

In principle, this identification requires validating a sequence of **public - key** certificates that terminates with the certificate of the originator. As noted in a previous subsection...

...but decrypted) form, along with the PEM header fields needed for signature verification (MIC-Info, **Originator - ID** -Asymmetric, **Originator** -Certificate, Issuer-Certificate). This form of storage is appropriate if the user wishes to forward...

...modification of the message while in storage. Finally, the user may save the message in **encrypted** form, to additionally protect the message against disclosure while in storage.

The Internet **Public - Key** Certification System

The PEM specifications encourage use of **public - key** cryptography for message integrity, **originator** authentication, and for distribution of data **encryption** keys. As noted previously, PEM makes use of **public - key** certificates that conform to CCITT X.509 (see the **Public Key** Certificates sidebar). The X.509 recommendation defines an authentication framework, not only certificate format, in...policies (e.g., they may strive for varying degrees of assurance in vouching for name- **public - key** bindings). However, X.509 makes no provisions for users to learn what policy each CA...

...imposed by these PCAs. MIT would achieve this capability by having two certificates, with different **public keys** , each signed by the relevant PCA.

In addition to the organizational CAs shown for MIT...end users, and thus is not addressed in this article.

(5) Any padding applied for **encryption** is removed as part of the decryption process performed by each recipient, so the padding...

...is employed for key distribution, the same general approach is employed. A different, symmetric cryptographic **key** is **shared** by each **originator - recipient** pair and that **key** is used to **encrypt** the message **key** on a per- **recipient** basis.

(7) The use of the recipient's distinguished name, rather than a mailbox name...

...a single cryptographically authenticated identity.

(8) This second field will generally be encoded, since the **encrypted** key is a binary value, and thus must be decoded before it can be decrypted ...

...Data Authentication, May 1985.

[9.] Federal information processing standards publication (FIPS PUB)

46-1, Data **encryption** standard, Reaffirmed Jan. 1988 (supersedes FIPS PUB 46, Jan. 1977).

[10.] Information processing systems--Open...

...framework for the discussion that follows.

A cryptoalgorithm is used to transform data through an **encryption** process. The Input to this process is called plaintext and the output is called ciphertext. The **encryption** is inverted by a decryption process, which accepts ciphertext as input and yields plaintext. In...

...is controlled by a key, which is a parameter to the process. In a symmetric (**secret - key**) cipher, the same key is used to **encrypt** and decrypt data. (In Figure A, KEY-1 and KEY-2 would be identical.) That **key** is kept **secret** and is shared by a transmitter and a receiver. Symmetric ciphers typically exhibit good performance and are used to **encrypt** user data. Privacy Enhanced Mall uses symmetric ciphers to **encrypt** messages.

In an asymmetric (**public - key**) cryptoalgorithm a pair of distinct, but mathematically related, keys are used for **encryption** and decryption. One **key** is kept **private** and is known only to its owner, whereas the other key is made publicly known, hence the term " **public - key** cryptography." (In Figure A, KEY-1 and KEY-2 would be distinct and either could be the **public** or **private** **key** .) Data **encrypted** with a user's **private** **key** can be decrypted using his **public** **key** , and vice versa, in the general model of **public - key** cryptography. Since the performance of asymmetric ciphers generally is not as good as that of...

...the former usually are not used to encipher user data directly. Instead, the asymmetry of **public - key** ciphers is often exploited to distribute symmetric keys and for digital signatures.

A digital signature is often effected using **public - key** cryptography and a (one-way) hash function, as illustrated in Figure B. The hash function...

...hash value, making it a "fingerprint" of the data. A user digitally signs data by **encrypting** the hash value of the data using his private component. Using an algorithm such as...the results. A match indicates a valid digital signature.

Privacy Enhanced Mail makes use of **public - key** ciphers to **encrypt** the symmetric key used to **encrypt** a message and to digitally sign a message.

Related article: **Public** **Key** Certificates: The X.509 Way

A **public - key** certificate is a data structure used to securely bind a **public** **key** to attributes. The attributes can consist of identification information, for example, a name, or authorization...

...certificates is contained within the international standards for directories. An X.509 certificate binds a **public** **key** to a directory name and identifies the entity who vouches for the binding. The whole...

...name of the entity that vouches for the binding between the subject name and the **public** **key** contained in the certificate (see the following subsection on distinguished names). The subject and issuer...

...certificate is valid, in much the same way as many credit cards are marked. The **public** **key** alluded to earlier, along with an identifier to specify the algorithm and any parameters required...

...the digital signature applied to the certificate. Both a one-way hash function and a **public - key** signature algorithm will be specified (e.g., the RSA **public key** algorithm and the MD2 hash algorithm in this example). This signature is applied by the...

...section "Cryptographic Concepts and Terminology." This use of certificates transforms the problem of acquiring the **public key** associated with a user into one of acquiring the **public key** of the issuer of the user's certificate. This issuer also will have a certificate ...

...The validation algorithm must conclude at some point, however, implying that the user holds a **public key** obtained through some out-of-band, integrity-secure channel (not through an untrusted network).
A...

...revoked ("hot-listed") by the issuer if the binding between the subject and the subject **public key** is no longer valid. This revocation of the binding may result from a number of...